

**Amendments to the Claims**

Please cancel Claims 2, 5, 7, 9-10, 14-16, 19-27, 29-30, 32, 34, 37-38, 41-42, 44-47, 50-53 and 56. Please amend Claims 1, 3, 6, 28, 31, 33, 40 and 43. The Claim Listing below will replace all prior versions of the claims in the application:

**Claim Listing**

1. (Currently amended) In a computer system, a method of retrieving sets of object data from a data store by defining queries on virtual relations which define desired sets of object data to be retrieved from [[a]] the data store, comprising the steps of:
  - providing a written representation of a desired object data set in terms of dimensions from relations storing object data in the data store and relation instances, the desired data set having a certain set type;
  - implying constraints on relation instances or dimensions including specifying sets of objects using constraints on dimensions and using a record operator to indicate which constraints apply to a same tuple in a relation, based on the set type of the desired data set and dimension expressions said step of implying constraints enabling length of the written representation to be minimized; and
  - using the written representation, querying to query the data store to retrieve and retrieving the desired object data set, including using the record operator to apply multiple atomic constraints on different dimensions in a same tuple in a relation, said querying and retrieving resulting in the desired object data set as query results.
2. (Cancelled)
3. (Currently amended) A method as claimed in Claim 1 wherein the step of providing a written representation includes using a record operator to express to a user where constraints on different dimensions apply to a same tuple in a relation or multiple constraints on a same dimension apply to multiple tuples ~~employing any combination of a~~

~~disjunctive expression and a conjunctive expression such that the written expression has one or more conjuncts; and~~

~~further comprising the steps of:~~

~~determining which relations to use in the evaluation of each conjunct in the written representation by using (i) the set type of the desired data set, (ii) a dimension list based on records in the conjunct of the written representation of the data set, and (iii) dimension relation associations and related inclusion criteria.~~

4. (Original) A method as claimed in Claim 1 wherein the step of providing a written representation includes employing any combination of a disjunctive expression and a conjunctive expression; and

further comprising the steps of:

performing OR-distribution on disjunctive expressions; and  
eliminating from disjunctive expressions, conjuncts with undefined binding variables.

5. (Cancelled)

6. (Currently amended) A method as claimed in Claim 1 ~~5~~ wherein further comprising the steps of:

translating a conjunctive expression ~~includes~~ including reordering terms of a conjunct;

~~further comprising the steps of:~~

moving terms with negation and no binding variables furthest back in the conjunct;

placing terms without negation ahead of other terms in the conjunct[[],];  
and

adding a terms with the certain set type of the desired data set to the front of the conjunct if all existing terms have negation.

7. (Cancelled)
8. (Original) A method as claimed in Claim 1 further comprising the step of automatically enforcing a record-operator where an expression in the written representation without the record-operator is semantically equivalent to the expression with the record-operator.
- 9-10. (Cancelled)
11. (Original) A method as claimed in Claim 1 wherein the data store has a native query engine; and  
further comprising the step of rewriting the written representation such that upon translation of the rewritten written representation into code for the native query engine, the code is optimized for querying the data store.
12. (Original) A method as claimed in Claim 1 wherein the step of providing a written representation includes utilizing a certain symbol to specify hierarchical constraints on dimensions.
13. (Original) A method as claimed in Claim 12 wherein the certain symbol is a colon or an equal sign followed by a colon.
- 14-16. (Cancelled)
17. (Original) A method as claimed in Claim 1 wherein the step of providing a written representation includes using an expression with bidirectional inlining of SQL statements and SDL statements; and  
further comprising the steps of  
denoting a start of SDL inlining in SQL with a bracket type symbol or  
referring to SDL metadata in an SQL statement by enclosing it in a bracket-operator.

18. (Original) A method as claimed in Claim 17 wherein the step of denoting includes denoting the start of SQL inlining in SDL with a square-bracket followed by a certain keyword  
or  
denoting the start of SQL inlining in SDL with a curly-bracket followed by a certain keyword.

19-27. (Cancelled)

28. (Currently amended) A method as claimed in Claim 1 further comprising the step of supporting federated queries by ~~any combination of: a) using prefixes or aliases to map dimension names and b) having session dependent federation in which metadata is defined at runtime and federation nicknames are generated on demand.~~

29-30. (Cancelled)

31. (Currently amended) In a computer system, apparatus for retrieving sets of object data from a data store by defining queries on virtual relations which define desired sets of object data to be retrieved from [[a]] the data store, comprising:  
an input component for providing a written representation of a desired object data set in terms of dimension[[s and]] attributes from relations storing object data in the data store, wherein the written representation specifies sets of objects using predicates on dimension attributes and employing a record operator notation to indicate which predicates apply to a same tuple; and  
an assembly coupled to receive the written representation, in response the assembly using the written representation to query the data store and retrieve the desired object data set by implying constraints on relations and determining number of dimensions and tuples in a relation to apply predicates, instances or dimensions by one of the set type of the desired data set and dimension expressions, said implying constraints

enabling length of the written representation to be minimized, the assembly producing an indication of the retrieved desired object data set as query results.

32. (Cancelled)
33. (Currently amended) Apparatus as claimed in Claim 31 wherein the written representation includes a record operator to express to a user which predicates on different dimensions apply to a same tuple in a relation and which multiple predicates on a same dimension apply to multiple tuples ~~any combination of disjunctive expressions and conjunctive expressions; and~~  
the assembly translates conjunctive expressions to respective SQL join terms and translates disjunctive expressions to respective SQL union terms.
34. (Cancelled)
35. (Original) Apparatus as claimed in Claim 33 wherein the assembly further reorders terms of a conjunctive expression.
36. (Original) Apparatus as claimed in Claim 31 wherein the assembly automatically enforces a record-operator where an expression in the written representation without the record-operator is equivalent to the expression with the record-operator.
- 37-38. (Cancelled)
39. (Original) Apparatus as claimed in Claim 31 wherein the written representation utilizes a certain symbol to specify hierarchical constraints on dimensions.
40. (Currently amended) Apparatus as claimed in Claim ~~[[31]]~~ 39 wherein the certain symbol is a colon or an equal sign followed by a colon.

41-42. (Cancelled)

43. (Currently amended) Apparatus as claimed in Claim 31 wherein the input component includes:

an editor for composing written representations; and

a data browser for enabling user browsing of dimension values and relations of the data store, to assist a user in composing desired written representations, wherein the editor employs a user interface which supports drag and drop of dimensions and relation values in written representations being composed.

44-47. (Cancelled)

48. (Original) Apparatus as claimed in Claim 31 wherein the written representation includes an expression with bidirectional inlining of SQL statements and SDL statements.

49. (Original) Apparatus as claimed in Claim 48 wherein said expression denotes SDL portions in an SQL statement with one bracket operator, and denotes SQL portions in an SDL statement with another bracket-type operator.

50-53. (Cancelled)

54. (Original) Apparatus as claimed in Claim 31 wherein the assembly supports federated queries of the data store by using prefixes or aliases to map dimension names.

55. (Original) Apparatus as claimed in Claim 54 wherein the assembly has session dependent federation in which metadata is defined at runtime and federation nicknames are generated on demand.

56. (Cancelled)